Persoonlijk plan

Functionaliteiten

Voor mijn individuele onderdeel ben ik van plan een Hill Climber-algoritme te maken met ook een simulated annealing toepassing. In het Hill Climber algoritme moet er een start state worden gemaakt die een valide of invalide oplossing is. Hierop wordt dan een random aanpassing gemaakt, waarna er wordt bekeken of de score vooruit is gegaan of niet. Is dit niet het geval, dan wordt de aanpassing ongedaan gemaakt. Is er wel sprake van een verbetering, dan kan er worden verder gegaan met de nieuwe state. Dit kan een bepaald aantal keer worden herhaald of totdat er geen verbetering meer nodig is.

Simulated Annealing werkt nagenoeg hetzelfde. Met dit algoritme wordt een verandering echter niet altijd ongedaan gemaakt wanneer de score niet verbeterd. In dit algoritme wordt de grootte van de kans berekend dat een verandering geaccepteerd wordt. Er kan dus voorkomen dat er een verslechtering wordt geaccepteerd.

Functies en classes

Voor deze algoritmes moeten er nog twee classes gemaakt worden. Zowel de Hill Climber als de Simulated Annealing hebben een eigen class. Dit zijn child-classes van de algorithm class en zij erven daarmee ook de methodes van deze class. De simulated Annealing class erft alle methodes van de Hill Climber class.

Hill Climber Class:

- Een methode die een geldige start state maakt om mee te beginnen. Het algoritme moet met zowel een volledig random (niet altijd valide) begin state (die kan worden gemaakt in een algorithm methode) als een begin state die een geldige oplossing is gebruikt kunnen worden. Deze begin state die een geldige oplossing is moet worden gemaakt in een methode die nog geschreven moet worden. Misschien kan deze methode uiteindelijk wel beter in de algorithm class geplaatst worden.
- Een methode die een random verandering maakt. Dit kan een route toevoegen of verwijderen of een connectie toevoegen of verwijderen zijn. Dit wordt random gekozen. Wanneer er een start state met een geldige oplossing wordt gebruikt moet de aanpassing die gemaakt wordt ook een nieuwe geldige oplossing maken. Hier kan misschien ook een aparte methode voor gemaakt worden. Dus een methode die een geldige random verandering maakt en een methode die een niet geldige random verandering maakt. Voor het maken van de veranderingen kunnen methodes uit het algoritme class worden aangeroepen.
- Een methode die de score van de nieuwe state vergelijkt met de score van de oude state.

- Een methode die de aanpassing weer ongedaan maakt en de state weer terugzet naar de oude state.
- Een methode die de state kan resetten en opnieuw kan beginnen.

Simulated Annealing

- Een methode die de kans berekent of een verandering doorgevoerd gaat worden.
- Een methode die op basis van de kans bepaalt of de verandering doorgevoerd gaat worden of niet en de state verandert naar de state waar het mee doorgaat.

Aansluiting op ander werk

Zoals hierboven beschreven gebruik ik in mijn algoritmes methodes uit de algorithm class. Daarbij kunnen ook de heuristieken gebruikt worden die hier ook in staan en die Gert gaat schrijven. Rayen gaat een plant propagation algoritme implementeren en kan daarvoor ook gebruik maken van mijn algoritme.

Anders uitgepakt

Voor het simulated annealing heb ik twee verschillende cooling schemes geïmplementeerd waarmee we kunnen experimenteren. Van tevoren wist ik niet dat dit een mogelijkheid was. Voor de hillclimber heb ik ook twee methodes gemaakt waarmee je verschillende soorten aanpassingen kan maken. We hebben een light versie en een heavy versie die ook routes kan verwijderen en toevoegen. Soms kwamen we er ook achter dat de classes toch niet altijd helemaal werkte zoals het hoorde of zoals er verwacht werd waardoor we soms heel lang bezig waren met het debuggen.

Commits

Datum	Link	Beschrijving
21-01-2024	https://github.com/Hachenberger02/AHRailNL/commit/11fb72a3a1931391384f75431559617f402b92f0	Begin aan het algoritme voor het maken van een random valide state.
22-01-2024	https://github.com/Hachenberger02/AHRailNL/commit/8b403afff0d23d40c7083a9be60ceb499a4db0b1	Het debuggen van de methode voor het algoritme voor het maken van een random valide state.
22-01-204	https://github.com/Hachenberger02/AHRailNL/commit/5e769b32691a62065a4aca9b1e280a96f6b62b8a	Code geschreven zodat gelijk een state wordt gemaakt en de while-loop in de methode voor het maken van een valide state geoptimaliseerd.
22-01-2024	https://github.com/Hachenberger02/AHRailNL/commit/34ac845d5945efa2aab3e505bd32d3189782d68a	Code toegevoegd voor het testen of de methode voor het maken van een valide state echt een valide state maakte.

22-01-2024	https://github.com/Hachenberger02/AHRailNL/commit/abf42577c7fe7f99f1dba014ce69c2cd2669575e	Code weggehaald voor het testen of de methode voor het maken van een valide state echt een valide state maakte
22-01-2024	https://github.com/Hachenberger02/AHRailNL/commit/1ee6db8596ca65f2d8dcddce4893fce49004f851	Assertions toegevoegd bij de methode voor het maken van een valide state.
23-01-2024	https://github.com/Hachenberger02/AHRailNL/commit/b9c695d05f04d56c32ab475383a61e6d0538164c	Functie gemaakt voor het toepassen van een random verandering voor de hill-climber.
25-01-2024	https://github.com/Hachenberger02/AHRailNL/commit/0d3f8994db6317cb5dec188af46087f72a33db64	Bugs opgelost waardoor hillclimber werkt.
25-01-2024	https://github.com/Hachenberger02/AHRailNL/commit/3f086e178284db7f0c8ce90282ad2e34dd8b69d0	Docstrings en comments toegevoegd aan hillclimber.
25-01-2024	https://github.com/Hachenberger02/AHRailNL/commit/8e619e1aa55dd3fb2c9b5cf51433e535eed5db25	Simulated annealing gemaakt en hillclimber geupdated.
25-01-2024	https://github.com/Hachenberger02/AHRailNL/commit/3f819fae01dbc652994e85b990a4f6a746ba9af7	Verdwenen code weer hersteld.
25-01-2024	https://github.com/Hachenberger02/AHRailNL/commit/711a31bb1432a0a9de3c3b8fe24be55a7e20f576	Fout bij hillclimber opgelost waardoor de route_index niet reset met het opnieuw runnen.
25-01-2024	https://github.com/Hachenberger02/AHRailNL/commit/1fabe4a6800b91d83d08c5b86ad4afb6f49c7c8f	Class toegevoegd voor een hillclimber restart.
26-01-2024	https://github.com/Hachenberger02/AHRailNL/commit/6c1a5987011d1e600a80fa96b6ba4ab2f39f9663	Hillclimber uitgebreid en docstrings en typehints toegevoegd bij simulated annealing
26-01-2024	https://github.com/Hachenberger02/AHRailNL/commit/75fa128c0deec508941a74882acf70ca7f03ac97	Comments toegevoegd bij simulated annealing